ss



**Sous le thème**

**Rapport de stage**

**5ème année**

**Ingénierie Informatique et Réseaux**

CONCEPTION ET RÉALISATION D’UN OUTIL D’AIDE À L’ANALYSE MÉTIER TÉLÉCOM

Période de stage

du 01 MARS 2023 au 30 AOÛT 2023

**Réalisé par :**

* JABRAN Ahmed
* ASRIH Ayoub

**Encadré par :**

Tuteur de l’école : Pr. ZERYOUH Meryam

Tuteur de stage : M. ELHADRI Badr

**Entreprise :** Atos IT Services

**Adresse :** Immeuble Suncity, 3ème étage, Rue Al Bortokal, Rabat

Remerciements

Le travail présenté dans ce rapport a été effectué dans le cadre du projet de fin d’études de la spécialité "MIAGE" à l'Ecole Marocaine de Sciences de l'Ingénieur Rabat.

Au terme de ce projet, Nous tenons à exprimer notre profonde gratitude et notre immense respect à madame ZERYOUH Meryam, enseignante chercheuse en informatique à l'école marocaine de sciences de l'ingénieur et à monsieur EL HADRI Badr, monsieur GHARBAOUI Yassine, monsieur HADIRI Aziz pour leurs disponibilités, leurs avis éclairés et leurs judicieux conseils.

Nous exprimons aussi notre gratitude à Mme. BENHIMA Assya, directrice adjoint des affaires pédagogiques, ainsi que l'ensemble de l'équipe de EMSI pour leur chaleureux accueil et aide.

Avec beaucoup d'égard, nous ne manquerons pas d'exprimer notre grande reconnaissance à tous les enseignants et administrateurs de l'école marocaine de sciences de l'ingénieur et toute l'équipe de l’entreprise Atos IT Services ainsi que les membres du jury pour avoir accepté de juger ce modeste travail.

Résumé

Ce rapport représente le résultat d'un travail effectué dans le cadre de notre stage de fin d'études au sein de l'entreprise Atos IT Services, dans le but d'obtenir notre diplôme d'Ingénieur d'État en Ingénierie Informatique et Réseaux à l'EMSI Rabat.

Notre projet s'est focalisé sur la conception et la réalisation d'un outil novateur spécialement dédié à l'analyse métier. Son objectif primordial était d'offrir une assistance précieuse à l'équipe de notre entreprise d'accueil dans l'analyse approfondie des besoins de ses clients. Afin d'atteindre cet objectif, nous avons développé deux applications distinctes, chacune jouant un rôle spécifique et complémentaire.

La première application est une application Spring Batch. Son rôle est de s'exécuter régulièrement afin de récupérer les documents à partir du serveur Jira, puis de les stocker et les indexer dans Elasticsearch.

La deuxième application est une application Angular qui offre aux utilisateurs une interface conviviale. Elle présente les données sous forme de tableau et intègre une fonctionnalité de recherche qui permet à l'utilisateur de consulter les documents en utilisant des mots-clés ou des tags, en fonction de son rôle spécifique.

**Mots clés:** Elasticsearch, Spring Batch, Angular, Serveur Jira, tags

Abstract

This report presents the results of our end-of-studies internship at Atos IT Services, aimed at obtaining our bachelor’s degree in computer and network Engineering from EMSI Rabat. Our project focused on designing and developing an innovative tool dedicated to business analysis. Its primary objective was to provide valuable assistance to our host company's team in conducting in-depth analysis of their clients' needs. To achieve this goal, we developed two distinct applications, each with a specific and complementary role.

The first application, a Spring Batch application, executes once a day to retrieve documents from the Jira server and store and index them in Elasticsearch. The second application, an Angular application, offers users a user-friendly interface. It presents data in a tabular format and incorporates a search functionality that allows users to access documents using keywords or tags, based on their specific role.

**Keywords:** Elasticsearch, Spring Batch, Angular, Jira Server, tags

Liste des abréviations

|  |  |
| --- | --- |
| Abréviations | Définition du terme |
| ORM | Object Relational Mappin |
| MVVM | Model View ViewMode |
| MVC | Model View Controller |
| REST | Representational State Transfer |
| CLI | Command Line Interface |
| JPA | Jakarta Persistence API |
| JDBC | Java Database Connectivity |
| IDE | Integrated Development Environment |
| HTTP | Hypertext Transfer Protocol |
| API | Application Programming Interface |
| URL | Uniform Resource Locator |
| SGBDR | Relational database management system |
| SQL | Structured Query Language |

Liste des figures

[Figure 1: Capture d’écran de l’outil Trello 6](#_Toc140587909)

[Figure 2 : Architecture applicative du système 11](#_Toc140587910)

[Figure 3 : Architecture Batch Processing 12](#_Toc140587911)

[Figure 4: Architecture MVVM 13](#_Toc140587912)

[Figure 5: Diagramme de cas d’utilisation utilisateur 14](#_Toc140587913)

[Figure 6 : Diagramme de cas d’utilisation de recherche 15](#_Toc140587914)

[Figure 7: Diagramme de cas d’utilisation Administrateur 16](#_Toc140587915)

[Figure 8 : Diagramme de classes 18](#_Toc140587916)

[Figure 9: Capture d’écran de l’outil Gitlab 20](#_Toc140587917)

[Figure 10: Capture d’écran de l’outil Workbench 21](#_Toc140587918)

[Figure 11: Capture d’écran de l’outil IntelliJ 22](#_Toc140587919)

[Figure 12: Capture d’écran de l’outil VSCode 23](#_Toc140587920)

[Figure 13: Capture d’écran de l’outil Postman 24](#_Toc140587921)

[Figure 14 : Capture d’écran Kibana 32](#_Toc140587922)

[Figure 15 : Base de données. 33](#_Toc140587923)

[Figure 16: Page d’authentification 34](#_Toc140587924)

[Figure 17: Page d’inscription 35](#_Toc140587925)

[Figure 18: Page d’accueil 36](#_Toc140587926)

Table des matières

[**Remerciements I**](#_Toc137919175)

[**Résumé II**](#_Toc137919176)

[**Abstract III**](#_Toc137919177)

[**Liste des abréviations IV**](#_Toc137919178)

[**Liste des figures V**](#_Toc137919179)

[**Introduction Générale 1**](#_Toc137919180)

[**Chapitre 1 : Contexte Général 2**](#_Toc137919181)

[**1.1 Présentation de l’organisme d’accueil 3**](#_Toc137919182)

[**1.1.1 Atos IT Services 3**](#_Toc137919183)

[**1.1.2 L'importance de l'analyse métier dans le domaine IT 3**](#_Toc137919184)

[**1.2 Etude de l’existant 4**](#_Toc137919185)

[**1.3 Présentation Générale du projet 4**](#_Toc137919186)

[**1.3.1 Exigences de projet 5**](#_Toc137919187)

[**1.3.2 Objectifs du projet 5**](#_Toc137919188)

[**1.4 Méthodologie de travail 5**](#_Toc137919189)

[**1.4.1 Outil de suivi Trello 5**](#_Toc137919190)

[**1.5 Conclusion 7**](#_Toc137919191)

[**Chapitre 2 : Analyse et conception 8**](#_Toc137919192)

[**2.1 Analyse des besoins 9**](#_Toc137919193)

[**2.1.1 Identification des acteurs 9**](#_Toc137919194)

[**2.1.2 Besoins fonctionnels 9**](#_Toc137919195)

[**2.1.3 Besoins non-fonctionnels 9**](#_Toc137919196)

[**2.1.4 Diagrammes des cas d’utilisation 10**](#_Toc137919197)

[**2.1.5 Diagrammes de classes 12**](#_Toc137919198)

[**2.2 Conception Générale 13**](#_Toc137919199)

[**2.2.1 Architecture applicative du système 13**](#_Toc137919200)

[**2.3 Conclusion 20**](#_Toc137919201)

[**Chapitre 3 : Implémentation de la solution 21**](#_Toc137919202)

[**3.1 Environnement logiciel 22**](#_Toc137919203)

Introduction Générale

Dans un contexte où la concurrence est de plus en plus rude, les entreprises cherchent à se différencier en offrant des produits et services de qualité supérieure à moindre coût. La gestion efficace des processus métiers est un facteur clé de succès dans ce contexte, et l'aide à l'analyse métier est un élément central de cette gestion efficace.

Dans ce cadre-là s’inscrit le sujet de notre PFE au sein de l’entreprise Atos IT Services, dont le but est de concevoir et implémenter une solution informatique avec une architecture moderne pour l’automatisation de l’analyse métier.

Le point de départ de notre projet est de faire une analyse profonde pour réaliser la première version, une réalisation qu’on peut la mettre en face des clients pour commencer à valider nos hypothèses, après on allons faire des améliorations correspondantes à nos besoins. L’équipe travaille avec une méthodologie Scrum selon les des sprints tracés dans la Roadmap du projet.

Le présent rapport décrit l’ensemble du travail réalisé dans le cadre de ce projet, il contient trois chapitres. Le premier chapitre contient une description du contexte général du projet notamment la présentation de l’entreprise Atos IT Services ainsi que la motivation et les objectifs du projet. Le deuxième chapitre présente une analyse de besoins fonctionnels et non fonctionnels en mettant l’accent sur l’ensemble des éléments de l’étude conceptuelle. Enfin, le chapitre quatre présentera les résultats de l’implémentation.

Chapitre 1 : Contexte général du projet

Ce chapitre abordera comme sujet la situation du contexte général du projet, sur un niveau organisationnel en présentant l’organisme d’accueil, et sur un niveau contextuel qui reflète la motivation et les objectifs du projet ainsi que la méthodologie de travail durant son déroulement.

## Présentation de l’organisme d’accueil

### Atos IT Services

Atos IT Services est une entreprise française de services du numérique, spécialisée dans la transformation digitale. Fondée en 1997, Atos IT Services est aujourd'hui présente dans plus de 70 pays, avec plus de 100 000 collaborateurs.

L'entreprise propose une large gamme de services et de solutions technologiques, allant de la cybersécurité à l'informatique en nuage en passant par l'analyse de données et la gestion de projets informatiques. Elle travaille avec des clients dans divers secteurs d'activité, notamment les services financiers, les télécommunications, les services publics, la santé, l'industrie et le commerce de détail.

Atos IT Services est également engagée dans la responsabilité sociale des entreprises, avec des initiatives pour réduire son empreinte carbone et encourager la diversité et l'inclusion sur le lieu de travail.

Grâce à son expertise technique et à son engagement en faveur de l'innovation et de la transformation numérique, Atos IT Services est bien placée pour aider les entreprises à relever les défis de l'ère numérique et à prospérer dans un environnement de plus en plus concurrentiel.

### L'importance de l'analyse métier dans le domaine IT

L'analyse métier est essentielle dans le domaine de l'informatique pour plusieurs raisons. Tout d'abord, les entreprises IT génèrent de grandes quantités de données et ces données doivent être analysées pour obtenir des informations exploitables. L'analyse métier permet aux entreprises de comprendre les tendances, les modèles et les relations entre les données. Cela peut aider les entreprises à prendre des décisions stratégiques éclairées et à améliorer leur efficacité opérationnelle.

En outre, l'analyse métier peut aider les entreprises IT à identifier les problèmes potentiels et à prendre des mesures proactives pour les résoudre. Par exemple, les analystes métier peuvent analyser les données de performance des applications pour identifier les goulots d'étranglement et les temps d'arrêt. Cela peut aider les entreprises à améliorer la qualité de leur service et à éviter les pertes financières dues aux temps d'arrêt.

Enfin, l'analyse métier peut aider les entreprises IT à mieux comprendre leurs clients. En analysant les données de comportement des utilisateurs, les entreprises peuvent identifier les préférences des clients, les tendances d'achat et les domaines d'amélioration des produits et services. Cela peut aider les entreprises à mieux cibler leur marketing et à améliorer la satisfaction client.

Dans l'ensemble, l'analyse métier est un élément clé de la réussite des entreprises IT. En utilisant des outils d'aide à l'analyse métier, les entreprises peuvent améliorer leur productivité, leur efficacité opérationnelle et leur compréhension des besoins des clients.

## Présentation Générale du projet

Notre projet de conception et réalisation d'un outil d'aide à l'analyse métier au sein de l'entreprise Atos IT Services vise à répondre aux besoins de staff Atos en leur offrant une solution informatique innovante pour faciliter la recherche et l'analyse dans les fichiers de spécifications. L'objectif principal de ce projet est de créer une application Spring Boot qui récupère des fichiers spécifications à partir d'un serveur web Jira. Ces fichiers seront ensuite indexés sous Elasticsearch pour faciliter leur recherche et leur analyse. Ensuite, les données seront affichées dans une application Angular, qui permettra aux utilisateurs de rechercher et d'analyser les fichiers spécifications de manière efficace et rapide. Cette solution permettra aux staff Atos IT Services d'optimiser leur temps et leurs ressources en facilitant leur travail quotidien d'analyse métier.

L’objectif générale du projet est de réaliser une application web qui sera utiliser par deux types des employés :

* **User :** dont le rôle est de bénéficier des ensembles de fonctionnalité fournis par notre application.
* **Administrateur :** dont le rôle est de gérer l’application.

### Problèmatique

----

### Objectifs du projet

Les exigences initiales indiquent que la solution sera une application web aidant les utilisateurs à :

* La recherche rapide par mots-clés ou bien par tag dans un grand nombre de fichiers spécifications.
* L‘automatisation d’indexation des documents dans le moteur de recherche Elastisearch.
* La consultation des derniers fichiers spécifications modifiés.

## Méthodologie de travail

### Outil de suivi Trello

Le Framework agile Scrum repose entre autres sur le principe de transparence. Certaines informations doivent donc être accessibles par tous, comme la tâche en cours de chacun, son état d’avancement, et l’objectif actuel de l’équipe. D’où l’importance que ces informations soient visibles en permanence.

C’est le tableau Scrum qui va jouer ce rôle. Il permet d’organiser le backlog, les tâches du sprint en cours et leur état d’avancement. Les tableaux Scrum peuvent être aussi simples qu’un tableau blanc et des posts-its, ou peuvent revêtir un format plus élaboré avec des logiciels spécialisés disposant de graphiques et de fonctionnalités de gestion des tâches plus avancées.

Pour notre tableau Scrum, nous utilisons Trello. Notre tableau est divisé en 5 listes qui correspondent au flux de travail des tâches :

* **À Faire :** quand on planifie notre sprint, on déplace les tâches du backlog vers cette liste.
* **En Cours** : contient les tâches en cours de développement et de réalisation.
* **Terminé** : la tâche est complète dans la phase du développement mais en attente de la validation fonctionnelle par le PO.
* **Approuvé :** une fois la tâche est approuvée fonctionnellement par le PO on peut la placer dans la colonne APPROUVÉ.
* **Bloqué :** on utilise cette liste lorsque la finalisation d’une tâche dépend d’un facteur externe, par exemple, on doit réaliser un achat et obtenir l’aval de mon PO, en spécifiant les raisons du blocage dans un commentaire.

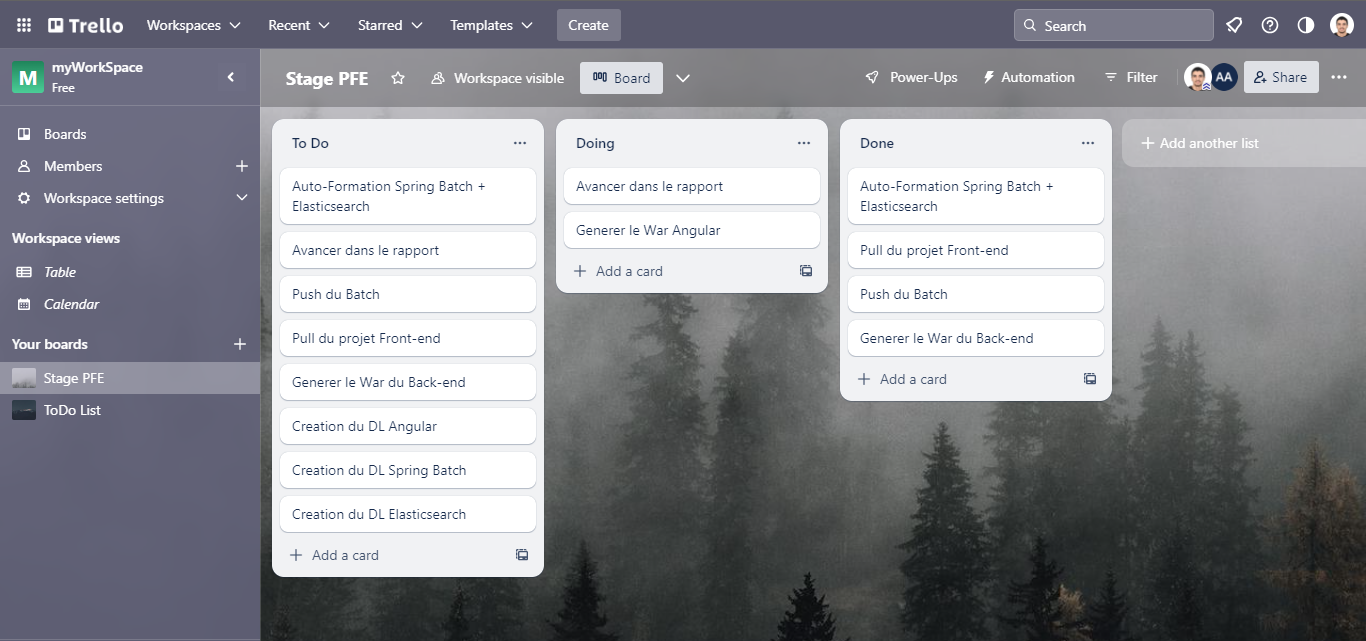


Figure 1: Capture d’écran de l’outil Trello

## Conclusion

Dans ce chapitre nous avons présenté l’ensemble des éléments qui permettent la situation de notre projet de stage dans son contexte organisationnel ainsi que la démarche de gestion du projet qui organise son déroulement et les outils utilisés.

Par la suite dans le chapitre suivant nous allons mettre l’accent sur l’étape de l’analyse et spécification des besoins qui permettra la collection des différents besoins afin de concevoir une solution qui répondra aux exigences exprimées.

Chapitre 2 : Analyse & Conception

Ce chapitre est consacré à l’analyse et à la spécification des besoins fonctionnels et non fonctionnels de la solution qui est une étape primordiale pour la réalisation de notre projet. Ainsi la conception générale du projet. Où nous allons passer à expliquer l’architecture globale du système qui représente une vue générale de notre solution.

## 2.1 Etude de l’existant

---

## 2.2 Analyse des besoins

Dans cette partie, nous présenterons les besoins fonctionnels et non fonctionnels identifiés après la sélection des besoins

### 2.2.1 Identification des acteurs

Dans le cas de notre projet on considère deux acteurs :

* **L’utilisateur :** l’acteur principale de notre application dont le rôle est de s’authentifier pour avoir l’accès aux différents cas d’utilisations.
* **L’admin :** en tant qu'acteur secondaire, joue un rôle clé dans notre application. Le rôle de l'administrateur est de gérer les utilisateurs, les autorisations et les paramètres de l'application.

### 2.2.2 Besoins fonctionnels

Au cours de cette étape, nous allons extraire les différentes fonctionnalités offertes par notre projet.

Notre application doit permettre à l’utilisateur de :

* S'inscrire/Se connecter en fournissant les informations nécessaires.
* Accéder à une liste de fichiers de spécifications disponibles.
* Afficher le contenue d'un fichier de spécifications sélectionné.
* Effectuer une recherche en entrant des mots-clés/tags pertinents.
* Afficher les résultats de recherche correspondants dans le tableau des données.
* Accéder à son profil utilisateur.
* Afficher les informations personnelles associées au profil.

### 2.2.3 Besoins non-fonctionnels

Outre les fonctions citées ci-dessus, l’application doit assurer en certaine mesure les caractéristiques suivantes :

* **L’efficacité :** L’efficacité de l’application doit permettre l’accès au page souhaité avec le minimum de manipulation et de temps.
* **La sécurité :** Les différents comptes utilisés par les utilisateurs doivent être sécurisés et vérifiés pour éviter les faux comptes et les fausses informations.
* **La fiabilité :** Touche à l’aspect qualité des données et persistance des informations dans l’application ainsi que la vitesse de chargement des interfaces.
* **La performance :** Le temps de réponse de la plateforme doit être rapide.
* **La maintenabilité :** La solution doit être stable face aux changements, ainsi qu’un fort niveau de testabilité assuré par les tests fonctionnels.
* **La scalabilité :** La solution doit d’être extensible en termes de la charge des requêtes traitées.
* **L’évolutivité :** Possibilité d’ajout des nouvelles fonctionnalités au cours du temps selon le besoin des utilisateurs.
* **Le déploiement intelligent :** L’introduction des nouveaux changements ne doit pas impacter les modules existants, d’où le besoin d’une démarche de déploiement intelligente.
* **La portabilité :** Facilité de passage d’un environnement de développement et tests vers un environnement de préproduction ou un environnement de production.

## 2.3 Architecture applicative du système

L’architecture de notre système est composée par deux parties :

* **Partie backend :** qui se représente par une application Spring Batch, dont le rôle est de s'exécuter régulièrement pour récupérer les documents à partir du serveur Jira, puis de les stocker et les indexer dans Elasticsearch. Pour atteindre cet objectif, l'application Spring Batch utilise une approche basée sur des tâches en lots (batch). Elle est configurée pour exécuter une série d'étapes définies, qui décrivent le flux de travail nécessaire pour traiter les documents provenant de Jira.
* **Partie frontend :** qui se représente par une application Angular offrant aux utilisateurs une interface conviviale et intuitive pour interagir avec notre système. Elle joue un rôle crucial dans la présentation des données aux utilisateurs de manière claire et organisée. Elle utilise des composants Angular pour structurer et afficher les informations de manière cohérente.

**A diagram of a diagram

Description automatically generated**

Figure 2 : Architecture applicative du système

### 2.3.1 Cas général pour le Backend

L'architecture utilisée dans le backend est celle du traitement par lots (Batch Processing Architecture). Cette architecture est spécialement conçue pour gérer des tâches de traitement en lots à grande échelle, telles que le traitement de volumes importants de données ou l'exécution de tâches planifiées en lot.

Elle comprend plusieurs composants clés, tels que les jobs, les étapes, les ItemReader, les ItemProcessor, les ItemWriter, le JobRepository et le JobLauncher. Ces éléments fonctionnent en harmonie pour exécuter des tâches de traitement par lots de manière efficace et scalable.

Le composant Job représente une unité de travail à exécuter et peut être composé de plusieurs étapes. Chaque étape définit une phase ou une tâche spécifique du processus de traitement par lots. L'ItemReader lit les données à partir d'une source donnée, tandis que l'ItemProcessor applique la logique métier ou les transformations sur les données lues. Ensuite, l'ItemWriter écrit les données traitées vers une destination spécifiée.

Le JobRepository est responsable de la gestion des métadonnées et de l'état des jobs en lot, tandis que le JobLauncher déclenche l'exécution des jobs en lot en fournissant les paramètres nécessaires.

A diagram of a work flow

Description automatically generated

Figure 3 : Architecture Batch Processing

### 2.3.2 Cas général pour le Frontend

L'architecture utilisée dans le frontend de notre application est celle du **Model-View-ViewModel** (**MVVM**). Cette architecture repose sur une approche interactive et axée sur les données pour la construction d'interfaces utilisateur dynamiques.

La **Vue** (View) définit l'interface utilisateur et la couche de présentation de l'application. Elle se concentre sur l'affichage des données et la capture des interactions de l'utilisateur. Dans notre cas, la Vue est créée à l'aide de modèles HTML combinés avec des directives et des liaisons Angular.

Le **ViewModel** (VueModèle) agit comme un intermédiaire entre le Model et la Vue. Il contient la logique de présentation de l'application, orchestre la liaison des données entre la Vue et le Model, et expose des propriétés et des commandes avec lesquelles la Vue peut interagir. Dans notre application, les composants Angular sont utilisés pour implémenter le ViewModel, encapsulant ainsi la Vue et la logique associée.

****

Figure 4: Architecture MVVM

## 2.4 Conception Générale

Cette partie se base sur la conception globale de l’architecture fonctionnelle de notre projet. Effectivement, elle explique de manière exhaustive les composants de notre système ainsi la communication entre eux.

### 2.4.1 Diagrammes des cas d’utilisation

Dans cette phase, on va traiter les différents cas d’utilisation et identifier les différentes interactions entre le système et les acteurs identifiés.

#### L’utilisateur :

La figure suivante représente le diagramme de cas d'utilisation globale de l'utilisateur qui permet de visualiser les différentes interactions entre l'utilisateur et le système, en identifiant les fonctionnalités principales offertes par le système du point de vue de l'utilisateur :

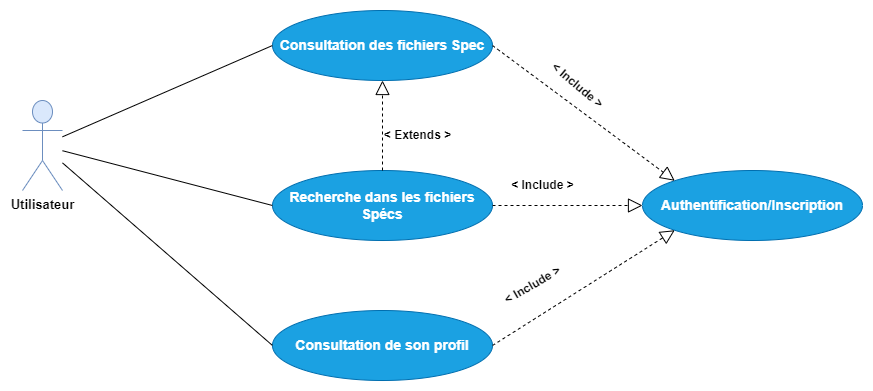


Figure 5: Diagramme de cas d’utilisation utilisateur

|  |  |
| --- | --- |
| Cas d’utilisation | Description |
| Consultation des fichiers spécs | Cela signifie que les utilisateurs peuvent consulter les fichiers de spécifications grâce à une table qui comprend les colonnes suivantes : problème, nom du fichier, dernière mise à jour et lien. La table présente une vue organisée des fichiers de spécifications, permettant aux utilisateurs de visualiser les problèmes associés, les noms des fichiers, les dates de dernière mise à jour et les liens pertinents vers les fichiers. Cette présentation structurée facilite la recherche et la consultation des fichiers de spécifications spécifiques, en fournissant aux utilisateurs les informations essentielles pour accéder aux fichiers pertinents et suivre les mises à jour les plus récentes. |
| Recherche dans les fichiers spécs | Les utilisateurs ont la possibilité d'effectuer des recherches dans les fichiers de spécifications. Ce cas d'utilisation permet aux utilisateurs de saisir des mots-clés ou des expressions dans un champ de recherche dédié, puis de lancer la recherche. Elasticsearch examine ensuite les fichiers de spécifications disponibles à la recherche de correspondances avec les termes saisis par l'utilisateur. Une fois la recherche effectuée, les résultats sont présentés à l'utilisateur, lui permettant de visualiser les fichiers de spécifications contenant les correspondances trouvées. |
| Consultation de son profil | L'utilisateur peut consulter les informations associées à son profil, telles que son nom, son adresse e-mail, son rôle dans le système, et d'autres détails pertinents. |
| Authentification/Inscription | L'utilisateur peut saisir ses informations d'identification, telles que son nom d'utilisateur et son mot de passe, pour se connecter au système. En cas de première utilisation, l'utilisateur peut également s'inscrire en fournissant les informations requises, telles que son nom, son adresse e-mail et un mot de passe. |

La figure suivante représente le cas d'utilisation de la recherche dans les fichiers spécs, mettant en évidence les interactions principales entre l'utilisateur et le système lorsqu'il effectue des recherches :

A picture containing text, diagram, line, screenshot

Description automatically generated

Figure 6 : Diagramme de cas d’utilisation de recherche

|  |  |
| --- | --- |
| Cas d’utilisation | Description |
| Recherche par mots-clés | Permet aux utilisateurs de saisir des mots-clés ou des expressions dans le champ de recherche dédié. Elasticsearch examine ensuite les fichiers de spécifications à la recherche de correspondances avec les termes saisis, et présente les résultats pertinents à l'utilisateur. |
| Recherche par tags | Permet aux utilisateurs de sélectionner des tags prédéfinis ou personnalisés pour affiner leur recherche. Les fichiers de spécifications sont étiquetés avec des tags correspondant à différents domaines, fonctionnalités ou catégories. En sélectionnant les tags pertinents, les utilisateurs peuvent restreindre les résultats de recherche aux fichiers étiquetés correspondants. |

#### L’administrateur :

La figure suivante représente le diagramme de cas d'utilisation globale de l’administrateur qui permet de visualiser les différentes interactions entre l’administrateur et le système, en identifiant les fonctionnalités principales offertes par le système du point de vue de l’administrateur :

A diagram of a flowchart

Description automatically generated with low confidence

Figure 7: Diagramme de cas d’utilisation Administrateur

|  |  |
| --- | --- |
| Cas d’utilisation | Description |
| Consultation de l’application | L'administrateur peut accéder à l'interface de l'application et parcourir les différentes fonctionnalités et informations disponibles. L'administrateur peut accéder à l'interface de l'application pour prendre connaissance de son contenu, examiner les différentes fonctionnalités et configurations disponibles, et gérer les paramètres du système en conséquence. |
| Gestion des utilisateurs | L'administrateur peut effectuer des actions telles que la création de nouveaux utilisateurs, la modification des informations utilisateur, la suppression des utilisateurs existants et la gestion des autorisations d'accès. Cette fonctionnalité offre à l'administrateur un contrôle complet sur la gestion des comptes utilisateur de l'application. |
| Gestion des droits d’accès | L'administrateur peut définir et ajuster les autorisations et les niveaux d'accès des utilisateurs en fonction de leurs rôles et responsabilités. Cette fonctionnalité offre à l'administrateur un contrôle précis sur les fonctionnalités et les données auxquelles les utilisateurs ont accès, assurant ainsi la sécurité et la confidentialité des informations de l'application. |
| Consultation de son Dashboard | La Dashboard administrateur affiche des informations clés telles que les statistiques d'utilisation, les rapports de performance et les notifications système importantes. Cela permet à l'administrateur de suivre et de surveiller efficacement l'état global de l'application. |

### 2.4.2 Diagrammes de classes

Pour développer l'architecture modulaire de l'application, nous allons nous concentrer sur l'identification des classes, des attributs et des opérations en nous basant sur le diagramme de classes fourni.

Le diagramme représente la structure de classes du système :

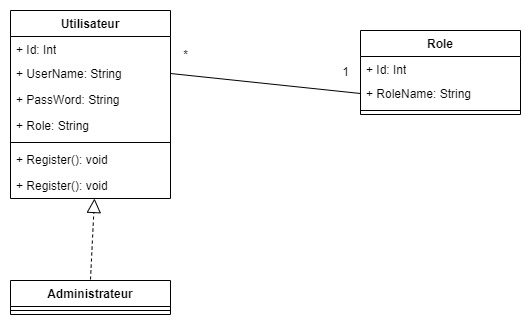


Figure 8 : Diagramme de classes

## 2.5 Conclusion

Dans ce chapitre nous avons détecté des besoins fonctionnels et non fonctionnels qui sont un complément de l’existant. Par la suite nous avons poussé l’analyse des besoins vers les diagrammes de cas d’utilisation et le diagramme de classes afin de visualiser les différentes fonctionnalités de la solution et les modules nécessaires nous avons entamé aussi la conception générale du projet en décrivant son architecture de manière détaillée.

Pour les technologies utilisées et les outils de développements, nous allons les présenter dans le prochain chapitre.

Chapitre 3 : Implémentation & Test

Ce chapitre aborde comme sujet les choix technologiques et les outils pour l’implémentation de notre solution ainsi que les captures d’écran des différentes IHM réalisées et les tests de validation effectués sur les modules développés.

## 3.1 Environnement logiciel

Tout au long de notre stage, nous avons utilisé une combinaison de technologies et d'outils modernes afin de développer et de déployer efficacement le projet :

* **Gitlab :** GitLab est une plateforme de gestion du cycle de vie du développement logiciel basée sur Git. Il offre un ensemble complet d'outils pour la gestion du code source, le contrôle de version, la collaboration, l'intégration continue, le déploiement continu et la gestion de projets.

En tant qu'outil de contrôle de version, GitLab permet de suivre les modifications apportées au code source et de gérer les différentes versions du projet. Il facilite la création de branches pour travailler sur des fonctionnalités ou des corrections de bugs spécifiques, puis fusionner ces branches dans la version principale du code.

GitLab propose également des fonctionnalités de collaboration, telles que les demandes de fusion (merge requests), qui permettent aux membres de l'équipe de passer en revue le code, de fournir des commentaires et de suggérer des modifications. Cela favorise un processus de développement collaboratif et garantit la qualité du code.

A screenshot of a computer

Description automatically generated

Figure 9: Capture d’écran de l’outil Gitlab

* **Workbench :** Workbench est un outil logiciel ou une plateforme qui fournit un environnement de développement intégré (IDE) aux développeurs pour créer, gérer et tester des applications logicielles. Elle offre un ensemble complet de fonctionnalités pour rationaliser le processus de développement de logiciels et augmenter la productivité des développeurs.

Workbench comprend généralement des outils pour écrire, éditer et déboguer du code, ainsi que des fonctionnalités de contrôle de version, de gestion de projet et de collaboration. Elle fournit un espace de travail centralisé où les développeurs peuvent écrire du code, compiler et exécuter des applications, et analyser leurs performances. Les workbenches prennent souvent en charge plusieurs langages de programmation et frameworks, permettant aux développeurs de travailler sur des projets divers en utilisant un seul outil.

L'objectif principal d'une workbench est de simplifier et d'optimiser le flux de travail de développement de logiciels en fournissant un environnement unifié qui intègre divers outils et fonctionnalités. Elle élimine le besoin de configuration manuelle et de mise en place, automatise les tâches répétitives et offre une gamme de fonctionnalités améliorant la productivité, telles que l'autocomplétion du code, le refactoring et l'analyse du code. En tirant parti d'une workbench, les développeurs peuvent se concentrer davantage sur la création des fonctionnalités principales de leurs applications et moins sur la configuration et la gestion des outils de développement.

A screenshot of a computer

Description automatically generated

Figure 10: Capture d’écran de l’outil Workbench

* **IntelliJ :** IntelliJ IDEA est un environnement de développement intégré (IDE) puissant et populaire utilisé par de nombreux développeurs Java. Il fournit une multitude de fonctionnalités et d'outils avancés qui améliorent la productivité et simplifient le processus de développement.

IntelliJ IDEA offre une interface conviviale et intuitive qui facilite la création, la modification et la gestion de projets Java. Il prend en charge une large gamme de technologies et de frameworks, y compris Spring Boot, ce qui en fait un choix courant pour le développement d'applications Spring.

L'IDE offre des fonctionnalités avancées d'édition de code telles que l'autocomplétion intelligente, la navigation rapide dans le code, la refactorisation et la détection d'erreurs en temps réel. Ces fonctionnalités aident les développeurs à écrire du code de manière plus rapide, efficace et sans erreurs.

IntelliJ IDEA intègre également des outils de débogage avancés qui permettent aux développeurs de détecter et de résoudre les problèmes plus facilement. Il prend en charge le débogage à distance, les points d'arrêt, l'inspection des variables, les évaluations d'expressions et bien plus encore.

A screenshot of a computer program

Description automatically generated

Figure 11: Capture d’écran de l’outil IntelliJ

* **VSCode :** Visual Studio Code (VS Code) est un éditeur de code source léger et très populaire développé par Microsoft. Il est largement utilisé par les développeurs pour différentes langues de programmation, y compris Java.

VS Code offre une interface conviviale et personnalisable. Il est doté d'une large gamme de fonctionnalités qui améliorent l'efficacité et la productivité des développeurs. Parmi ces fonctionnalités, on retrouve :

-**Édition de code avancée :** VS Code propose une coloration syntaxique, une mise en forme automatique, une indentation intelligente, des suggestions de code et une navigation rapide. Il prend également en charge des fonctionnalités spécifiques à Java, telles que la recherche de définitions, l'accès aux références et la documentation intégrée.

- **Extensions et intégration avec des outils :** VS Code dispose d'une vaste bibliothèque d'extensions créées par la communauté qui étendent ses fonctionnalités. Il prend en charge des extensions Java spécifiques, telles que les outils de génération de code, les frameworks de test et les intégrations avec des outils de build comme Maven et Gradle.

**- Débogage et tests :** VS Code intègre des fonctionnalités de débogage pour Java, ce qui permet aux développeurs de placer des points d'arrêt, d'exécuter le code pas à pas et d'inspecter les variables en temps réel. Il prend également en charge l'exécution de tests unitaires et fournit des rapports détaillés sur les résultats des tests.

**-** **Intégration avec les systèmes de contrôle de version :** VS Code est compatible avec les principaux systèmes de contrôle de version tels que Git. Il offre une interface conviviale pour gérer les branches, les commits et les conflits, ainsi que pour effectuer des opérations courantes telles que le push et le pull.

A screenshot of a computer program

Description automatically generated

Figure 12: Capture d’écran de l’outil VSCode

* **Postman :** Postman est un outil de collaboration et de développement d'API largement utilisé par les développeurs. Il permet de tester, de déboguer et de documenter facilement les API, facilitant ainsi le processus de développement d'applications basées sur les API.

Avec Postman, vous pouvez envoyer des requêtes HTTP et observer les réponses pour tester et valider les endpoints de votre API. Il offre une interface conviviale où vous pouvez spécifier les paramètres, les en-têtes, les corps de requête et les méthodes HTTP pour effectuer différentes opérations (GET, POST, PUT, DELETE, etc.).

Postman permet également d'organiser et de sauvegarder vos requêtes dans des collections. Vous pouvez créer des scénarios de tests enchaînant plusieurs requêtes, et les exécuter automatiquement pour vérifier le bon fonctionnement de votre API.

A screenshot of a computer

Description automatically generated

Figure 13: Capture d’écran de l’outil Postman

## 3.2 Architecture technique de système

### 3.2.1 Couche Backend

Pour la partie backend de notre projet, nous avons opté pour l'utilisation des technologies suivantes :

* **Spring Batch :** L'architecture Spring Batch est basée sur un modèle de traitement par lots également connu sous le nom de **Batch Processing**. Elle fournit un ensemble de composants et de fonctionnalités pour faciliter le développement et l'exécution de traitements par lots dans les applications Java. Voici les principaux composants de l'architecture Spring Batch :
* **Job :**

- Un "job" représente un processus global à exécuter, souvent composé de plusieurs étapes.

- Il est défini à l'aide de configurations XML ou de code Java.

- Un job peut être exécuté de manière planifiée ou déclenchée manuellement.

* **Step :**

- Une "étape" est une unité de traitement distincte au sein d'un job.

- Chaque étape effectue une tâche spécifique, comme la lecture des données, le traitement des données et l'écriture des résultats.

- Une étape peut être configurée pour traiter les données en parallèle ou en séquentiel.

* **ItemReader :**

- L'interface "ItemReader" définit la méthode de lecture des données à partir d'une source de données.

- Il peut lire des données à partir de différentes sources telles que des fichiers plats, des bases de données, des services web, etc.

* **ItemProcessor :**

- L'interface "ItemProcessor" définit la méthode de traitement des données lues par l'ItemReader.

- Il permet de transformer ou de filtrer les données avant de les écrire.

* **ItemWriter :**

- L'interface "ItemWriter" définit la méthode d'écriture des données traitées par l'ItemProcessor.

- Il peut écrire les données dans différentes cibles, telles que des fichiers, des bases de données, des services web, etc.

* **JobRepository :**

- Le "JobRepository" est responsable de la gestion des métadonnées des jobs et des étapes exécutées.

- Il stocke les informations sur l'état d'exécution des jobs et permet de reprendre un job à partir du point d'arrêt en cas de panne.

* **JobLauncher :**

- Le "JobLauncher" est utilisé pour lancer l'exécution d'un job.

- Il récupère les configurations du job à partir du "JobRepository" et coordonne l'exécution des étapes du job.

* **Spring Boot :** Spring Boot est un framework Java qui simplifie et accélère le processus de développement d'applications basées sur Spring. Voici les principaux composants et fonctionnalités de Spring Boot :
* **Autoconfiguration :** Spring Boot offre une fonctionnalité d'autoconfiguration qui permet de configurer automatiquement de nombreux aspects de l'application. Il détecte les bibliothèques présentes dans le classpath et configure automatiquement les composants en fonction de ces bibliothèques. Cela permet de réduire la configuration manuelle et d'accélérer le développement en adoptant des conventions par défaut.
* **Dépendances starters :** Les starters de Spring Boot sont des dépendances préconfigurées qui simplifient l'intégration de technologies spécifiques dans une application. Ils fournissent un ensemble cohérent de dépendances, de configurations et de valeurs par défaut pour une technologie donnée. Par exemple, le starter "spring-boot-starter-web" inclut les dépendances nécessaires pour développer des applications web avec Spring MVC.
* **Serveur d'application embarqué :** Spring Boot intègre un serveur d'application embarqué (tel que Tomcat, Jetty ou Undertow) qui permet d'exécuter l'application directement, sans nécessiter de configuration externe. Cela facilite le déploiement de l'application, car il n'est pas nécessaire de déployer l'application sur un serveur d'application séparé.
* **Actuator :** Spring Boot Actuator fournit des fonctionnalités de surveillance et de gestion de l'application. Il expose des endpoints **REST** qui permettent de surveiller les métriques, les informations sur la santé de l'application, les statistiques de performance, etc. Actuator facilite le suivi et la gestion des applications en production.
* **Spring Boot CLI :** La CLI (Command-Line Interface) de Spring Boot est un outil en ligne de commande qui permet de développer rapidement des applications Spring Boot. Elle offre une expérience de développement fluide en permettant de créer, de tester et de lancer des applications directement depuis la ligne de commande.
* **Intégration avec les technologies populaires :** Spring Boot facilite l'intégration avec de nombreuses technologies couramment utilisées, telles que les bases de données relationnelles (JPA, JDBC), les bases de données NoSQL, la messagerie, la sécurité, les services web, etc. Il fournit des starters et des configurations par défaut pour simplifier l'intégration de ces technologies.

### 3.2.2 Couche Frontend

Pour la partie backend de notre projet, nous avons choisi d'utiliser les technologies suivantes :

* **Angular :** Angular est un framework de développement d'applications web développé par Google. Il utilise le langage TypeScript pour créer des applications web réactives et dynamiques. Voici les principaux composants de l'architecture Angular :
* **Modules :**

- Les "modules" sont des containers qui regroupent des fonctionnalités étroitement liées.

- Chaque application Angular est constituée d'au moins un module racine.

- Les modules déclarent les composants, les services et les autres fonctionnalités utilisés dans l'application.

* **Composants :**

- Les "composants" sont les éléments de construction de l'interface utilisateur dans Angular.

- Chaque composant encapsule un morceau d'interface utilisateur avec son propre modèle, ses styles et sa logique.

- Les composants interagissent les uns avec les autres pour créer une application web riche en fonctionnalités.

* **Templates :**

- Les "templates" sont des fichiers HTML qui définissent la structure et l'apparence des composants.

- Les templates incluent des balises HTML standard ainsi que des directives Angular pour rendre le contenu dynamique.

- Les templates sont associés aux composants et sont utilisés pour générer l'interface utilisateur de l'application.

* **Directives :**

- Les "directives" sont des instructions spéciales qui modifient le comportement des éléments HTML.

- Angular fournit des directives intégrées telles que \*ngFor, \*ngIf, etc., qui permettent de manipuler dynamiquement le contenu des templates.

- Les développeurs peuvent également créer leurs propres directives personnalisées pour étendre les fonctionnalités d'Angular.

* **Services :**

- Les "services" sont des classes réutilisables qui fournissent des fonctionnalités spécifiques à l'application.

- Les services gèrent les opérations de récupération des données, les appels HTTP, la logique métier, etc.

- Ils sont injectés dans les composants et d'autres services pour partager des fonctionnalités communes.

* **Routage :**

- Le "routage" permet la navigation entre les différentes vues de l'application.

- Le routage est défini dans les modules et permet d'associer des URL spécifiques à des composants.

- Il facilite la navigation de page en page sans recharger complètement l'application.

* **Dépendances :**

- Angular gère les dépendances entre les différents composants et services.

- Les dépendances sont injectées via l'injection de dépendances, ce qui facilite la gestion des relations entre les différentes parties de l'application.

* **Angular Material :** Angular Material est une bibliothèque de composants UI prête à l'emploi développée par l'équipe Angular de Google. Elle offre une collection complète de composants et de directives pour créer des interfaces utilisateur modernes et attractives dans les applications Angular. Voici les principaux composants et fonctionnalités d'Angular Material :
* **Composants UI :** Angular Material fournit une large gamme de composants prêts à l'emploi tels que des boutons, des cartes, des formulaires, des tables, des menus, des listes, des dialogues, des onglets, des barres de progression, etc. Ces composants sont conçus selon les directives de conception Material Design de Google et sont hautement personnalisables pour s'adapter aux besoins de l'application.
* **Thèmes et styles :** Angular Material offre un système de thèmes et de styles pour une personnalisation aisée de l'apparence des composants. Les thèmes prédéfinis et personnalisables permettent de définir rapidement une palette de couleurs cohérente pour l'application. Les styles prédéfinis et les classes CSS facilitent la mise en forme des composants selon les besoins spécifiques de l'interface utilisateur.
* **Responsivité :** Les composants d'Angular Material sont conçus pour être réactifs et s'adapter automatiquement à différents appareils et tailles d'écran. Ils fournissent des fonctionnalités intégrées de mise en page flexible, de grille adaptable et de disposition réactive pour garantir une expérience utilisateur optimale sur tous les types d'appareils.
* **Accessibilité :** Angular Material accorde une grande importance à l'accessibilité. Les composants sont conçus en tenant compte des bonnes pratiques d'accessibilité, tels que la prise en charge des technologies d'assistance, la gestion des couleurs contrastées et l'utilisation de balises sémantiques appropriées. Cela garantit que les applications développées avec Angular Material sont accessibles à tous les utilisateurs.
* **Animation :** Angular Material propose des fonctionnalités d'animation intégrées pour ajouter des transitions et des effets visuels fluides aux composants. Il prend en charge les animations CSS et les animations basées sur JavaScript, ce qui permet de créer des interfaces utilisateur dynamiques et engageantes.
* **Intégration avec Angular :** Angular Material est étroitement intégré à Angular, ce qui facilite son utilisation dans les projets Angular existants. Il peut être installé et importé facilement à l'aide des outils de gestion des dépendances tels que npm. Les composants d'Angular Material sont également compatibles avec les fonctionnalités d'Angular telles que la gestion des formulaires, la validation, l'injection de dépendances, etc

### 3.2.3 Couche Base de données

Pour la partie backend de notre projet, nous avons choisi d'utiliser les technologies suivantes :

* **Elasticsearch :** Elasticsearch est un moteur de recherche et d'analyse de données distribué basé sur Apache Lucene. Il est conçu pour fournir une recherche rapide et des capacités d'analyse avancées sur de grandes quantités de données. Voici les principaux composants de l'architecture Elasticsearch :
* **Cluster :**

- Un "cluster" est un ensemble de nœuds Elasticsearch qui coopèrent pour stocker et traiter les données.

- Chaque cluster est identifié par un nom unique.

- Les données sont réparties entre les nœuds du cluster pour permettre la haute disponibilité et la scalabilité.

* **Nœud :**

- Un "nœud" est une instance d'exécution d'Elasticsearch.

- Chaque nœud participe au cluster en gérant les données et en exécutant les opérations de recherche et d'indexation.

- Les nœuds peuvent être configurés pour différents rôles, tels que nœud de données, nœud de maître ou nœud de coordination.

* **Index :**

- Un "index" est une collection logique de documents similaires.

- Les documents sont regroupés dans des index en fonction de leur structure et de leur objectif.

- Chaque index a un nom unique et peut être configuré avec des paramètres spécifiques, tels que le nombre de réplicas et le nombre de fragments.

* **Document :**

- Un "document" est l'unité de base de stockage dans Elasticsearch.

- Il représente une entrée de données JSON pouvant être indexée et recherchée.

- Chaque document est associé à un ID unique et appartient à un index spécifique.

* **Shard :**

- Un "shard" est une partie d'un index contenant une fraction des données.

- Les shards permettent de répartir les données et les opérations de recherche/écriture entre les nœuds du cluster.

- Elasticsearch gère automatiquement la répartition des shards pour garantir une répartition équilibrée.

* **Réplica :**

- Un "réplica" est une copie exacte d'un shard primaire.

- Les réplicas sont créés pour fournir une redondance et une tolérance aux pannes.

- Chaque shard primaire peut avoir plusieurs réplicas.

* **Recherche :**

- Elasticsearch offre une recherche plein texte avancée et des fonctionnalités de recherche facettée.

- Vous pouvez interroger les données à l'aide de requêtes structurées, de filtres, de recherche en texte intégral et d'agrégations pour obtenir des résultats précis.

* **API :**

- Elasticsearch expose une API RESTful qui permet d'interagir avec le cluster.

- Vous pouvez utiliser des requêtes HTTP pour effectuer des opérations d'indexation, de recherche, de mise à jour et de suppression des documents.

A screenshot of a computer

Description automatically generated

Figure 14 : Capture d’écran Kibana

* **MySQL :** MySQL est un système de gestion de base de données relationnelle (SGBDR) largement utilisé dans le développement d'applications. Il offre une architecture solide et des fonctionnalités avancées pour stocker, gérer et interroger les données de manière efficace. Voici quelques points clés à connaître sur MySQL :
* **Architecture client-serveur :** MySQL suit une architecture client-serveur, où le serveur MySQL gère les connexions, les requêtes et la gestion des données, tandis que les clients se connectent au serveur pour accéder aux bases de données.
* **Langage SQL** : MySQL utilise le langage SQL (Structured Query Language) pour interagir avec les données. SQL permet d'exécuter des requêtes pour récupérer, insérer, mettre à jour et supprimer des données, ainsi que pour créer et modifier la structure des bases de données et des tables.
* **Types de données :** MySQL prend en charge divers types de données, tels que les entiers, les décimales, les chaînes de caractères, les dates, les heures, les booléens, etc. Il offre également des types de données spécifiques, tels que les types géospatiaux pour la manipulation des données géographiques.
* **Tables et relations :** MySQL organise les données dans des tables, qui sont des structures de données rectangulaires composées de colonnes et de lignes. Les tables sont utilisées pour représenter des entités et leurs relations dans une base de données relationnelle. Les relations entre les tables sont établies à l'aide de clés primaires et de clés étrangères pour maintenir l'intégrité référentielle.
* **Indexation :** MySQL utilise des index pour accélérer les opérations de recherche dans les tables. Les index créent des structures de données supplémentaires qui permettent de trouver rapidement les enregistrements correspondant à certains critères de recherche. Les index peuvent être créés sur une ou plusieurs colonnes pour améliorer les performances des requêtes.
* **Fonctions et procédures stockées :** MySQL prend en charge les fonctions et procédures stockées, qui sont des blocs de code SQL pouvant être enregistrés et réutilisés. Les fonctions retournent une valeur, tandis que les procédures effectuent des opérations sans renvoyer de valeur. Cela permet de simplifier et de réutiliser la logique métier côté serveur.

A screenshot of a computer

Description automatically generated

Figure 15 : Base de données.

## 3.3 Implémentation & test

Tout au long de cette partie, on va traiter notre solution comme 1ère version, puis on va donner des captures d’écran du résultat courant de notre projet avec les tests de validation.

### 3.3.1 Description générale de la solution

Notre Application est répartie en deux phases. Dans la première phase, nous récupérons les données du serveur Jira à l'aide d'un processus en lots (batch) et nous les indexons dans Elasticsearch. La deuxième phase consiste à récupérer les données d'Elasticsearch et à les afficher dans Angular, offrant ainsi à l'utilisateur la possibilité de les visualiser de différentes manières.

### 3.3.2 Capture d’écran du résultat courant

* 1. **Authentification**

Cette capture d'écran présente les interfaces d'authentification de l'application. L'utilisateur est invité à saisir ses informations d'identification, telles que son nom d'utilisateur et son mot de passe, avant de pouvoir accéder à l'ensemble des fonctionnalités de l'application. Cette page d'authentification garantit la sécurité en vérifiant l'identité de l'utilisateur avant de le connecter à l'application. L'interface est conçue de manière conviviale, avec des champs clairement étiquetés et des indications claires sur les informations requises. Un bouton de connexion est également fourni pour valider les informations saisies et permettre à l'utilisateur de continuer à utiliser l'application.

A screenshot of a computer

Description automatically generated

Figure 16: Page d’authentification

De plus, cette fonctionnalité d'authentification est rendue possible grâce à l'utilisation d'AuthGuard dans Angular. AuthGuard est un mécanisme intégré à Angular qui permet de gérer les autorisations d'accès aux routes de l'application. Ainsi, si l'utilisateur n'est pas connecté, il sera automatiquement redirigé vers la page de connexion et ne pourra pas accéder à la page d'accueil ni à aucune autre fonctionnalité de l'application. Cela garantit une protection supplémentaire en empêchant l'accès non autorisé aux fonctionnalités de l'application.

* 1. **Inscription**

La deuxième capture d'écran présente l'interface d'inscription de l'application. Les utilisateurs sont invités à remplir les champs requis tels que le nom d'utilisateur, le mot de passe, l'adresse e-mail et le rôle. Cette interface facilite le processus d'inscription en fournissant des étiquettes claires pour chaque champ et des indications précises sur les informations requises. Une fois que l'utilisateur a saisi toutes les informations nécessaires, il peut valider son inscription en cliquant sur le bouton "Register". Une fois l'inscription réussie, l'utilisateur peut utiliser les informations d'identification fournies lors du processus d'inscription pour se connecter à l'application et accéder à toutes les fonctionnalités disponibles.

A screenshot of a computer

Description automatically generated

Figure 17: Page d’inscription

De plus, si l'utilisateur possède déjà un compte, il lui est proposé de se rendre sur la page de connexion en cliquant sur le bouton "New User ? Register Now !". Cela permet aux utilisateurs existants de passer directement à la phase de connexion sans avoir à remplir à nouveau les informations d'inscription. Cette fonctionnalité offre une expérience pratique pour les utilisateurs qui souhaitent simplement accéder à l'application plutôt que de s'inscrire à nouveau.

* 1. **Page d’accueil**

La deuxième capture d'écran présente l'interface d'inscription de l'application. Les utilisateurs sont invités à remplir les champs requis tels que le nom d'utilisateur, le mot de passe, l'adresse e-mail et le rôle. Cette interface facilite le processus d'inscription en fournissant des étiquettes claires pour chaque champ et des indications précises sur les informations requises. Une fois que l'utilisateur a saisi toutes les informations nécessaires, il peut valider son inscription en cliquant sur le bouton "Register". Une fois l'inscription réussie, l'utilisateur peut utiliser les informations d'identification fournies lors du processus d'inscription pour se connecter à l'application et accéder à toutes les fonctionnalités disponibles.

A screenshot of a computer

Description automatically generated

Figure 18: Page d’accueil

De plus, si l'utilisateur possède déjà un compte, il lui est proposé de se rendre sur la page de connexion en cliquant sur le bouton "New User ? Register Now !". Cela permet aux utilisateurs existants de passer directement à la phase de connexion sans avoir à remplir à nouveau les informations d'inscription. Cette fonctionnalité offre une expérience pratique pour les utilisateurs qui souhaitent simplement accéder à l'application plutôt que de s'inscrire à nouveau.

## 3.4 Conclusion

À la fin de ce chapitre, nous avons réussi à expliquer l’architecture techniques de notre système, l’état d’avancement de notre projet ainsi sa mise en œuvre.

Conclusion Générale

En conclusion, ce projet de PFE au sein de l'entreprise Atos IT Services a été une opportunité enrichissante pour étudier et mettre en œuvre une solution informatique innovante dans le domaine de l'analyse métier télécom. En travaillant en étroite collaboration avec une méthodologie Scrum et en suivant une approche itérative, notre équipe a réussi à réaliser une première version fonctionnelle de la solution, permettant ainsi la validation de nos hypothèses.

Au cours de ce projet, nous avons également identifié et analysé les besoins fonctionnels et non fonctionnels, en mettant l'accent sur une étude conceptuelle approfondie. En tirant parti de nouvelles technologies, frameworks et outils, nous avons pu améliorer nos compétences techniques et professionnelles, en particulier en travaillant avec Elasticsearch. Cette expérience nous a permis de renforcer notre adaptabilité aux nouvelles exigences et de surmonter les défis rencontrés tout au long du projet.

Sur le plan personnel, ce stage a été une occasion précieuse de développer nos compétences en communication, de renforcer notre capacité à travailler en équipe et d'établir des liens avec des experts du domaine. Nous sommes reconnaissants envers Atos IT Services pour cette opportunité qui nous a permis de découvrir le monde professionnel et d'améliorer nos capacités à tous les niveaux.

En somme, ce projet de PFE a été une étape importante dans notre parcours académique et professionnel. Il nous a permis d'approfondir nos connaissances, d'acquérir de l'expérience pratique et de nous préparer à relever de nouveaux défis dans le domaine de la gestion efficace des processus métiers. Nous sommes fiers des résultats obtenus et sommes convaincus que cette solution informatique contribuera à l'amélioration des performances de l'entreprise Atos IT Services dans un marché concurrentiel.

Bibliographie

Atos:

<https://atos.net/fr/>

Elasticsearch:

<https://www.elastic.co/what-is/elasticsearch>

Spring Boot:

<https://www.ibm.com/topics/java-spring-boot#:~:text=Java%20Spring%20Boot%20(Spring%20Boot,ability%20to%20create%20standalone%20applications>

MySQL:

<https://www.techtarget.com/searchoracle/definition/MySQL>

Angular:

<https://angular.io/guide/what-is-angular#:~:text=Angular%20is%20a%20development%20platform,client%2Dserver%20communication%2C%20and%20more>

Angular Material:

<https://www.webagesolutions.com/blog/introduction-to-angular-material#:~:text=Angular%20Material%20components%20are%20Built,Material%20components%20speeds%20up%20development>

Architecture Elasticsearch:

<https://medium.com/geekculture/elasticsearch-architecture-1f40b93da719>

Architecture Spring Batch:

<https://terasoluna-batch.github.io/guideline/5.0.0.RELEASE/en/Ch02_SpringBatchArchitecture.html>

Architecture Angular:

<https://angular.io/guide/architecture>

Architecture MVVM:

<https://www.geeksforgeeks.org/mvvm-model-view-viewmodel-architecture-pattern-in-android/>